WHAT IS CLAIMED IS:

1. An apparatus comprising a set of computer instructions stored on a computer readable medium and executable by a computer processor to:

receive a filesystem request indicating a requested filesystem operation;

compare the filesystem request to a programmable rulesbase to determine whether the filesystem request matches a pattern; and

if the filesystem request matches a pattern, execute an action associated with the pattern.

2. The apparatus of Claim 1, wherein the set of computer instructions are further executable to:

perform the requested filesystem operation; and

execute the action out-of-band with the requested filesystem operation.

3. The apparatus of Claim 1, wherein the set of computer instructions are further executable to:

execute the action in-band;

determine whether to proceed with the requested filesystem operation; and

perform the requested filesystem operation if it is determined that the requested operation should proceed.

4. The apparatus of Claim 3, wherein the determination of whether to perform the requested filesystem operation is based on a result of the action.

5. The apparatus of Claim 3, wherein a result of the performance of the requested filesystem operation is affected by the result of the action.

6. The apparatus of Claim 1, wherein the computer readable instructions are further executable to compare a set of data affected by the requested file system operation to the pattern to determine if the filesystem request matches the pattern.

7. An apparatus comprising a set of computer executable instructions stored on a computer readable medium, the set of computer instructions comprising:

a filesystem manager operable to:

maintain a filesystem for an underlying storage medium;

receive a filesystem request from a filesystem client that includes a requested filesystem operation;

pass the filesystem request to a rules engine; and

determine whether to proceed with the requested filesystem operation;

a programmable rules base comprising a set of rules, wherein each rule further comprises a pattern and an associated action; and

a rules engine operable to:

receive the filesystem request;

compare the filesystem request to the one more rules in the programmable rules base to determine if the filesystem request matches one or more rules from the set of rules based on the pattern for each rule;

if there are one or more matching patterns, execute the action associated with each matching pattern;

return a rules engine response to the filesystem manager.

8. The apparatus of Claim 7, wherein the filesystem manager determines to perform the requested operation based on the rules engine response.

9. The apparatus of Claim 7, wherein the rules engine executes at least one action associated with the one or more matching patterns in-band.

10. The apparatus of Claim 9, wherein the rules engine executes at least one action associated with the one or more matching patterns out-of-band.

11. A method for applying rules to filesystem operations comprising:

receiving a filesystem request indicating a requested filesystem operation;

comparing the filesystem request to a programmable rulesbase to determine the filesystem request matches a pattern; and

if the filesystem request matches a pattern, executing an action associated with the pattern.

12. The method of Claim 11, further comprising:

performing the requested filesystem operation; and

execute the action out-of-band with the requested filesystem operation.

13. The method of Claim 11, further comprising:

executing the action in-band;

determining whether to proceed with the requested filesystem operation; and

performing the requested filesystem operation if it determined that the requested operation should proceed.

14. The method of Claim 3, wherein the determination of whether to perform the requested filesystem operation is based on a result of the action.

15. The method of Claim 3, wherein a result of the performance of the requested filesystem operation is affected by the result of the action.

16. The apparatus of Claim 11, further comprising comparing a set of data affected by the requested file system operation to the pattern to determine if the filesystem request matches the pattern.

17. An apparatus comprising a set of computer instructions stored on a computer readable medium and executable by a computer processor, the set of computer instructions comprising instructions executable to:

receive a requested operation;

gather a predefined set of metadata based on the requested operation;

group the predefined set of metadata according to an attribute to generate one or more groups of metadata; and

apply predefined heuristics to at least one of the groups of metadata to generate a statistic of interest.

18. The apparatus of Claim 17, wherein the set of computer instructions further comprises instructions executable to:

determine which pieces of the predefined metadata of interest in each of the one or more groups has been impacted by the application of the predefined heuristics; and

purge a particular group of metadata when each of the pieces of metadata of interest in the particular group has been impacted by the application of the predefined heuristics.

19. The apparatus of Claim 18, wherein the set of computer instructions further comprise instructions executable to gather the predefined metadata synchronously with performance of the requested operation.

20. The apparatus of Claim 18, wherein the set of computer instructions further comprise instructions executable to gather

the predefined metadata asynchronously with performance of the requested operation.

21. A method of managing a statistical warehouse comprising:

receiving a requested operation;

gathering a predefined set of metadata based on the requested operation;

grouping the predefined set of metadata according to an attribute to generate one or more groups of metadata; and

applying predefined heuristics to at least one of the groups of metadata to generate a statistic of interest.

22. The method of Claim 21, further comprising:

determining which pieces of the predefined metadata of interest in each of the one or more groups has been impacted by the application of the predefined heuristics; and

purging a particular group of metadata when each of the pieces of metadata of interest in the particular group has been impacted by the application of the predefined heuristics.

23. The method of Claim 22, further comprising:

gathering the predefined metadata synchronously with performance of the requested operation.

24. The method of Claim 22, further comprising gathering the predefined metadata synchronously with performance of the requested operation.

25. An apparatus comprising a set of computer instructions stored on a computer readable medium and executable by a computer processor, the set of computer instructions comprising instructions executable to:

receive a requested operation;

gather a predefined set of metadata based on the requested operation;

store the set of predefined metadata as a set of records corresponding to one or more tables, wherein each record contains a piece of metadata of interest and wherein each record is associated with a group of one or more groups;

apply predefined heuristics to at least a portion of the set of records associated with a particular group to generate a statistic of interest.

26. The apparatus of Claim 25, wherein the set of computer instructions further comprise instructions executable to:

associate a list of table names to the particular group based on the grouping attribute; and

determine when a particular table corresponding to a particular table name in the list of table names has been impacted by one or more of the predefined heuristics based on a list of records impacted by the one or more predefined heuristics.

27. The apparatus of Claim 26, wherein the set of computer instructions further comprise instructions executable to:

determine when all the tables corresponding to all the table names in the list of table names have been impacted by the one or more of the predefined heuristics; and

when all the tables corresponding to all the table names in the list of table names have been impacted by the one or more of the predefined heuristics, purge all records corresponding to the particular group.

28. The apparatus of Claim 27, wherein the set of computer instructions further comprise instructions executable to associate each record with a group of one or more groups based on when the predefined metadata in each record was gathered.

29. The apparatus of Claim 28, wherein the set of computer instructions further comprise instructions executable to gather the predefined metadata synchronously with performance of the requested operation.

30. The apparatus of Claim 28, wherein the set of computer instructions further comprise instructions executable to gather the predefined metadata asynchronously with performance of the requested operation.

31. A method of managing a statistical warehouse, comprising:

receiving a requested operation;

gathering a predefined set of metadata based on the requested operation;

storing the set of predefined metadata as a set of records corresponding to one or more tables, wherein each record contains a piece of metadata of interest and wherein each record is associated with a group of one or more groups;

applying predefined heuristics to at least a portion of the set of records associated with a particular group to generate a statistic of interest.

32. The method of Claim 31, further comprising:

associating a list of table names to the particular group based on the grouping attribute; and

determining when a particular table corresponding to a particular table name in the list of table names has been impacted by one or more of the predefined heuristics based on a list of records impacted by the one or more predefined heuristics.

33. The method of Claim 32, further comprising:

determining when all the tables corresponding to all the table names in the list of table names have been impacted by the one or more of the predefined heuristics; and

when all the tables corresponding to all the table names in the list of table names have been impacted by the one or more of the predefined heuristics, purging all records corresponding to the particular group.

34. The method of Claim 33, further comprising associating each record with a group of one or more groups based on when the predefined metadata in each record was gathered.

35. The method of Claim 34, further comprising gathering the predefined metadata synchronously with performance of the requested operation.

36. The method of Claim 34, further comprising gathering the predefined metadata asynchronously with performance of the requested operation.

37. A system comprising:

a network;

a plurality of client computers, each client computer comprising:

a client processor;

a client network interface to connect to and interface with the network;

a client computer readable medium accessible by the client processor, storing a client program executable by the client processor to:

generate a first filesystem request;

receive a first filesystem response;

an intermediary device comprising:

an intermediary processor;

an intermediary network interface to connect to and interface with the network;

an intermediary computer readable medium accessible by the intermediary processor and executable to:

provide a client-facing filesystem interface;

provide a server-facing filesystem interface;

receive the first filesystem request from a requesting client according to the client-facing filesystem interface;

pass the first filesystem request to a server as a proxy request according to the server-facing filesystem interface;

receive a server response from the server according to the server facing interface;

pass the server response to the requesting client as the first filesystem response;

a plurality of servers, each server further comprising:

a server processor;

a server interface coupled to the server processor to connect to and interface with the network;

a server computer readable medium storing a server program executable by the server processor to:

provide an origin filesystem;

receive the proxy request from the intermediary device;

execute a requested operation;

generate the server response; and

communicate the server response to the intermediary computer.

38. The system of Claim 37, wherein the intermediary program is further executable to modify the first filesystem request to generate the proxy request.

39. The system of Claim 37, wherein the intermediary program is executable to apply active rules to the first filesystem request.

40. The system of Claim 37, wherein the intermediary program is further executable to modify the server response to generate the first filesystem response.

41. The system of Claim 37, wherein the intermediary program is further executable to:

determine whether to service the first filesystem request;

if the intermediary program determines that it should not service the first filesystem request, generate a redirect response;

communicate the redirect response to the requesting client.

42. The system of Claim 41, wherein the client program at each client is further executable to:

receive the redirect response as the first filesystem response;

generate a second filesystem request; and

communicate the second filesystem request to the origin server.

43. The system of Claim 40, wherein the server program at the origin server is further executable to:

receive the second filesystem request;

execute a requested operation;

generate a second server response; and

pass the second server response to the requesting client.

44. The system of Claim 37, wherein the intermediary program is further executable to:

define an import space comprising one or more of the origin filesystems;

define an export space comprising one or more export filesystems;

wherein the one or more export filesystems are based on the one or more origin filesystems in the import space.

45. The system of Claim 44, wherein the intermediary program is further executable to stack organize the one or more origin filesystems in the import space into a stack.

46. The system of Claim 45, wherein the intermediary program is further executable to stack organize the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories in higher origin filesystems in the stack.

47. The system of Claim 45, wherein the first filesystem request further comprises the requested operation and a file upon which the requested operation is to occur.

48. The system of Claim 47, wherein the intermediary program is further executable to pass the request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

49. The system of Claim 47, wherein the intermediary program is further executable to pass the request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

50. The system of Claim 47, wherein the intermediary program is further executable to flag a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

51. The system of Claim 50, wherein the particular file and particular other files share a common name.

52. The apparatus of Claim 37, wherein the intermediary program is further executable to:

compare the first filesystem request to a programmable rulesbase to determine if the first filesystem request matches a pattern; and

if the filesystem request matches a pattern, execute an action associated with the pattern.

53. The system of Claim 52, wherein the intermediary program is further executable to execute the action out-of-band.

54. The system of Claim 52, wherein the intermediary program is further executable to execute the action in-band.

55. An intermediary device comprising:

a processor;

a network interface to connect to and interface with a network;

a computer readable medium accessible by the processor and executable to:

provide a client-facing filesystem interface;

provide a server-facing filesystem interface;

receive a filesystem request from a requesting client according to the client-facing filesystem interface;

pass the filesystem request to a server as a proxy request according to the server-facing filesystem interface;

receive a server response from the server according to the server-facing interface; and

pass the server response to the requesting client as a proxy response.

56. The device of Claim 55, wherein the intermediary program is further executable to:

determine whether to further process the filesystem request;

generate a redirect reply;

communicate the redirect reply to the requesting client.

57. The device of Claim 56, wherein the redirect reply is configured to prompt the requesting client to generate a second filesystem request to the server.

58. The system of Claim 55, wherein the intermediary program is further executable to modify the filesystem request to generate the proxy request.

59. The system of Claim 55, wherein the intermediary program is further executable to modify the server response to generate the proxy response.

60. The device of Claim 55, wherein the intermediary program is further executable to:

present a union filesystem via the client-facing interface.

61. The device of Claim 60, wherein the intermediary program is further executable to:

define an import space comprising one or more of the origin filesystems;

define an export space comprising one or more export filesystems;

wherein the one or more export filesystems are based on the one or more origin filesystems in the import space.

62. The device of Claim 61, wherein the intermediary program is further executable to stack organize the one or more origin filesystems in the import space into a stack.

63. The device of Claim 62, wherein the intermediary program is further executable to stack organize the one or more origin filesystems by subsuming files and directories from lower

origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack.

64. The device Claim 62, wherein the filesystem request further comprises the requested operation and a file upon which the requested operation is to occur.

65. The device of Claim 64, wherein the intermediary program is further executable to pass the proxy request based on the filesystem request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

66. The device of Claim 64, wherein the intermediary program is further executable to pass the proxy request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

67. The device of Claim 64, wherein the intermediary program is further executable to flag a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

68. The device of Claim 67, wherein the particular file and particular other files share a common name.

69. The device of Claim 55, wherein the intermediary program is further executable to:

compare the filesystem request to a programmable rulebase to determine if the filesystem request matches a pattern; and

if the filesystem request matches a pattern, execute an action associated with the pattern.

70. The device of Claim 69, wherein the intermediary program is further executable to execute the action out-of-band.

71. The device of Claim 69, wherein the intermediary program is further executable to execute the action in-band.

72. A device comprising a set of computer instructions stored on a computer readable medium executable by a computer processor, aid set of computer instructions comprising instructions executable to:

provide a client-facing filesystem interface;

provide a server-facing filesystem interface;

receive a filesystem request from a requesting client according to the client-facing filesystem interface;

pass the filesystem request to a server as a proxy request according to the server-facing filesystem interface;

receive a server response from the server according to the server-facing interface; and

pass the server response to the requesting client as a proxy response.

73. The device of Claim 72, wherein the set of computer instructions further comprise instructions executable to:

determine whether to further process the filesystem request;

generate a redirect reply;

communicate the redirect reply to the requesting client.

74. The device of Claim 73, wherein the redirect reply is configured to prompt the requesting client to generate a second filesystem request to the server.

75. The system of Claim 72, wherein the set of computer instructions further comprise instructions executable to modify the filesystem request to generate the proxy request.

76. The system of Claim 72, wherein the set of computer instructions further comprise instructions executable to modify the server response to generate the proxy response.

77. The device of Claim 72, wherein the set of computer instructions further comprise instructions executable to:

present a union filesystem via the client-facing interface.

78. The device of Claim 77, wherein the set of computer instructions further comprise instructions executable to:

define an import space comprising one or more of the origin filesystems;

define an export space comprising one or more union filesystems;

wherein the one or more union filesystems are based on the one or more origin filesystems in the import space.

79. The device of Claim 78, wherein the set of computer instructions further comprise instructions executable to stack organize the one or more origin filesystems in the import space into a stack.

80. The device of Claim 79, wherein the intermediary program is further executable to stack organize the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack.

81. The device Claim 79, wherein the filesystem request further comprises the requested operation and a file upon which the requested operation is to occur.

82. The device of Claim 81, wherein the set of computer instructions further comprise instructions executable to pass the proxy request based on the filesystem request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

83. The device of Claim 81, wherein the set of computer instructions further comprise instructions executable to pass the proxy request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

84. The device of Claim 81, wherein the set of computer instructions further comprise instructions executable to flag a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

85. The device of Claim 84, wherein the particular file and particular other files share a common name.

86. The device of Claim 72, wherein the intermediary program is further executable to:

compare the filesystem request to a programmable rulebase to determine if the filesystem request matches a pattern; and if the filesystem request matches a pattern, execute an action associated with the pattern.

87. The device of Claim 86, wherein the intermediary program is further executable to execute the action out-of-band.

88. The device of Claim 86, wherein the intermediary program is further executable to execute the action in-band.

89. A method of managing a filesystem comprising:

providing a client-facing filesystem interface;

providing a server-facing filesystem interface;

receiving a filesystem request from a requesting client according to the client-facing filesystem interface;

passing the filesystem request to a server as a proxy request according to the server-facing filesystem interface;

receiving a server response from the server according to the server-facing interface; and

passing the server response to the requesting client as a proxy response.

90. The method of Claim 89, further comprising:

determining whether to further process the filesystem request;

generating a redirect reply;

communicating the redirect reply to the requesting client.

91. The method of Claim 90, wherein the redirect reply is configured to prompt the requesting client to generate a second filesystem request to the server.

92. The method of Claim 89, further comprising modifying the filesystem request to generate the proxy request.

93. The method of Claim 89, further comprising modifying the server response to generate the proxy response.

94. The method of Claim 89, further comprising presenting a union filesystem via the client-facing interface.

95. The method of Claim 94, further comprising:

defining an import space comprising one or more of the origin filesystems;

defining an export space comprising one or more union filesystems;

wherein the one or more union filesystems are based on the one or more origin filesystems in the import space.

96. The method of Claim 95, wherein further comprising stack organizing the one or more origin filesystems in the import space into a stack.

97. The method of Claim 96, further comprising stack organizing the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack.

98. The method Claim 96, wherein the filesystem request further comprises the requested operation and a file upon which the requested operation is to occur.

99. The method of Claim 98, further comprising passing the proxy request based on the filesystem request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

100. The method of Claim 98, further comprising passing the proxy request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

101. The method of Claim 98, further comprising flagging a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

102. The method of Claim 101, wherein the particular file and particular other files share a common name.

103. The method of Claim 89, further comprising:

comparing the filesystem request to a programmable rulebase to determine if the filesystem request matches a pattern; and

if the filesystem request matches a pattern, executing an action associated with the pattern.

104. The method of Claim 103, further comprising executing the action out-of-band.

105. The method of Claim 103, further comprising executing the action in-band.

106. A device comprising a set of computer instructions stored on a computer readable medium executable by a processor, said set of computer instructions comprising instructions executable to: define an import space comprising one or more of the origin filesystems;

define an import space comprising one or more of the origin filesystems;

define an export space comprising one or more union filesystems;

wherein the one or more union filesystems are based on the one or more origin filesystems in the import space.

107. The device of Claim 106, wherein the set of computer instructions further comprise instructions executable to stack organize the one or more origin filesystems in the import space into a stack.

108. The device of Claim 107, wherein the intermediary program is further executable to stack organize the one or more origin filesystems by subsuming files and directories from lower origin filesystems in the stack into similarly named files and directories from higher origin filesystems in the stack.

109. The device Claim 107, wherein the computer instructions are further executable to receive a filesystem request comprising a requested operation and a file upon which the requested operation is to occur.

110. The device of Claim 109, wherein the set of computer instructions further comprise instructions executable to pass the filesystem request to a topmost origin filesystem in the stack that contains the file upon which the requested operation is to occur.

111. The device of Claim 109, wherein the set of computer instructions further comprise instructions executable to pass the filesystem request to a topmost origin filesystem in the stack that contains an innermost directory associated with the file upon which the requested operation is to occur.

112. The device of Claim 109, wherein the set of computer instructions further comprise instructions executable to flag a particular file in an upper origin filesystem in the stack to prevent particular other files in one or more lower origin filesystems in the stack from becoming visible.

113. The device of Claim 112, wherein the particular file and particular other files share a common name.